DOCUMENT RESUME

ED 156 206 IR 006 439

AUTHOR Berkey, Dennis D.
TITLE A Method for the Use of Computing in a Conventional Calculus Course.
PUB DATE Jun 77
NOTE 7p.; Paper presented at the Conference on Computers in the Undergraduate Curricula (East Lansing, Michigan, June 20-22, 1977); For entire proceedings, see IR 006 142

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS *Calculus; *College Mathematics; *Computer Assisted Instruction; Course Descriptions; *Course Evaluation; Higher Education; Instructional Improvement; *Integrated Curriculum; *Programing; Student Reaction; Teaching Techniques; Undergraduate Study

ABSTRACT

An experiment at Boston University was designed to demonstrate that the computer can be successfully integrated into the traditional introductory calculus course. Over half of the students in an introductory course in 1976 voluntarily participated in a programming project involving applications of the concepts to be studied during the semester. All work was to be done on the University's IBM 370/158 time sharing system, whose interactive terminals are available at many locations across the campus. Student response to the projects at the end of the semester was very positive. The projects also resulted in very little out-of-class demand on the instructor's time. (Author/CHV)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED156206

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

A METHOD FOR THE USE OF COMPUTING IN A CONVENTIONAL CALCULUS COURSE

BY

Dennis D. Berkey

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Ted Sjoersma

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) AND USERS OF THE ERIC SYSTEM."

IR006439

A Method for the Use of Computing in a Conventional Calculus Course

Dennis D. Berkey
Department of Mathematics
Boston University
264 Bay State Road
Boston, Massachusetts 02215
(617) 353-2560

Introduction

During the past decade the advantages of using the computer in teaching introductory calculus have been well-established, a good amount of relevant material has found its way into print ([12], [15], [16], [17], [9]) and the availability of computing facilities has become almost universal in this country. However, despite all these developments, the use of computers in introductory calculus instruction still has not found widespread adoption ([15]).

With respect to this question, the more radical departures seem to have received a majority of the attention, at least in the mathematical literature. The purpose of this article is to describe a way in which the use of computers can be successfully integrated into the confines of the traditional calculus course, and to report on one such implementation.

Which Approach?

Many different models for the use of computers in calculus have been proposed, with none having been established as a clear preference.

The most radical approach has probably been that embodied by the Center for Research in College Instruction in Science and Mathematics (CRICASAM) project [8] which called for the integration of computing into a completely restructured calculus course. A textbook was produced which placed major emphasis on algorithmic concepts, resulting in a true computer-based introduction of the calculus. Although initially pursued with much interest and vigor, this approach has not found major adoption.

Another common approach has been to establish a separate course in parallel with the calculus, usually entitled *Calculus and Computing*, or something similar, in which the basics of programming are introduced and the students write programs involving topics from the elementary calculus. These courses usually carry one or two credits, and the student is expected either to have already completed some study in the calculus, or to be doing so concurrently. As such, these courses amount to an additional course basically concerned with the concepts of the elementary calculus.

In contrast to the above approaches, some calculus instructors have made use of the computer in a much more passive way. Either the computer has been used as a demonstration device (by simply displaying output or, in some cases by bringing a mini-computer into the classroom [11] or else instructors have written and "canned" programs which students simply execute at time-sharing terminals [7]. Along the same lines some institutions have used their time-sharing capabilities to develop self-help laboratories for calculus students ([1], [3], [5], [14]). These approaches have the advantage of minimizing the time spent on computer work, among other advantages, but do not involve the student in an active way with the programs being used.

It seems clear at this point that a wholesale changeover to a computer-based approach to

the introductory calculus is not going to occur to any large extent. Moreover, in light of several related developments which have been occurring during the past decade, it is the opinion of this author that renewed attention should be given to the question of how best to integrate some nontrivial use of the computer into the traditional presentation of the elementary calculus without having to send students into separately designed laboratory or computer-calculus mini-courses.

First, students are now arriving much more sophisticated in the understanding and use of calculating devices than they were only a few years ago. Indeed, the advent of the inexpensive (sometimes programmable) hand calculator and the increased availability of computing facilities in the high schools have drastically closed the gap between the experience our students bring with them into the introductory mathematics courses and the understanding of the algorithmic methods necessary to successfully write and execute simple programs at a computing facility. (The article [9] contains an excellent up-to-date survey of the capabilities of hand calculators now generally available.)

Second, the accessibility of versatile high-level programming languages has significantly increased over the past decade. A few years ago, when one had to choose an initial programming language, a dichotomy existed between choosing a simple but limited instructional language (usually BASIC, or one locally produced) or a more versatile but cumbersome language such as FORTRAN for mathematical programming, or COBOL for business applications and manipulation of large data banks. With the development of PL/1 (and the PLC subset) one now has the opportunity of introducing students directly to a language which is versatile, yet simple to learn and use, and which they can continue to use should their interests in computing carry them beyond the type of programming done in their introductory calculus courses.

Third, and probably most important, is the rapid development of computer science as a separate discipline, in many ways distinct from mathematics. Indeed, undergraduate programs in computer science are well-developed at many institutions. Entry into most of these programs is not at the post-calculus point, but at the beginning of the freshman year. Moreover, the problems and goals set forth in these introductory courses often bear little semblence to the computational problems in calculus and numerical analysis through which we used to feel that we were introducing our students to "computing." (The "core" area of computer science education has been described in [13] as including the study of (1) programming languages, (2) operating systems, and (3) other subject matter concerned with the application of computing techniques to the study of computers.) Even at this author's institution, where the computer science programs, both undergraduate and graduate, have been developed wholly within the department of mathematics, it is not uncommon for an undergraduate to complete his studies with a concentration in computer science having been exposed to little of what is referred to as traditional mathematics, pure or applied. The point here is not a criticism of the emerging curricula in computer science—it is rather, to emphasize that in making use of the computer in the limited context of illustrating the implementation of the concepts of calculus we can no longer claim to be introducing students to computer science. That, for the most part, is something else.

In light of the above observations and the obvious fact that computers now play an important part in many of the related fields in which our students will eventually study and/or work (not to mention daily life) it is indeed desirable to make use of the computer in a nontrivial way in our calculus instruction. However, in doing so we need not feel obligated to devote any large amounts of time to introducing the "concepts of computing" or to make any significant changes in the traditional presentation of the calculus. Instead, we should simply introduce students to the facilities at hand, show them how to gain access to these facilities, and equip them with a few basic skills in writing the particular

type of programs sufficient to carry out the calculations at hand. It is the opinion of this author that this can be accomplished within the confines of the traditional introductory calculus course, in minimal time, without having to resort to separate courses or tutorials for this purpose. For those students whose interest in computing extends beyond this limited framework, an introductory course in computer science is the logical next step. For all students, however, it should be made clear that the computer is an easily-accessible device which can be put to good use without a long period of training on the part of the user.

The Experiment

To illustrate the author's claim that the computer can be successfully integrated into the traditional introductory calculus course the following experiment was conducted during the fall semester of 1976 at Boston University.

In an introductory course in the calculus, students were invited to participate in a programming project involving applications of the concepts to be studied during the semester. All work was to be done on the University's IBM 370/158 time sharing system whose interactive terminals are available at many locations across the campus.

One class period was initially devoted to an explanation of the computing facilities available to the students, the way by which one signs onto and off the computing system at a console, inputting and executing programs, and a very brief introduction to the PLC programming language. A sample program was presented which computed the mean of three numbers entered by the user via an interactive loop in the program. A short handout was distributed which described the procedures for signing on and off, and the use of the editing routine.

Once each week thereafter the students were given a one-page handout containing a printout of a program written by the instructor which performed a calculation relating to a topic currently under study in the course. In addition to the program the handout contained a set of instructions which typically asked the student to: (1) enter the program into his own program library, using the edit package to correct whatever mistakes were made during entry, (2) use the program to carry out several calculations (the programs were always written with an interactive loop which would ask the user to enter new data at the end of each calculation), (3) answer in writing several questions about what particular lines in the program were accomplishing, and (4) use the edit routine to modify the given program to one which would achieve a slightly different objective. (The objective, but not the particular modifications, was all that was given here.) About fifteen minutes of class time were devoted each week to a brief description of what the program was doing, new features not seen in previous programs, and student questions.

Thus, while the students received some initial information in writing each week (the sample program), the idea was that many of the basic ideas associated with the programs—looping, go-to statements, if statements, function routines—would be noticed and understood through interaction with working programs, rather than via the lecture method, as the complexity of the programs increased.

As reference material the students were advised to consult two publications available in the University bookstore. One [4] is a description of the University's computing facilities. The other [2] is an introductory text which happened to be in use in an introductory course in our computer science program.

The type of projects assigned were chosen from among the typical uses of the computer in illustrating the calculus—approximation of definite integrals, areas between curves and volumes, error analysis for Taylor series expansions, partial sums of infinite series, etc.—which have previously been well documented. The students were instructed to retain all printouts and to keep them, together with their written responses to the questions, in a notebook which was submitted at the end of the course.

Preparation time for these projects on the part of the instructor was found to be really minimal. Once a program was written and operating a listing was obtained on a ditto master. (For those who must work strictly with punched cards, a photo copy of the correct output could be used here.) Instructions were added, and the copies were run for distribution. The instructor, previously unfamiliar with the PL/1 language, found the text by Pollach and Sterling [10] an interesting and highly readable introduction to the language.

Results

Response to the projects was very positive. Of the sixty-five students enrolled in the course, thirty-seven elected to participate in the projects.

At the end of the course a questionnaire was distributed to these students concerning the projects, to which twenty-seven submitted responses. The following subjective questions were asked:

1. Were the projects effective in illustrating the concepts from the calculus on which they were based?

2. Should the approach have been varied towards either the use of strictly "canned" programs which students would merely run but not enter or modify, or, at the opposite extreme, the requirement that students write all their own programs "from scratch"?

3. In what ways can the projects be improved?

In response to question one, twenty-five students said yes, two no. The affirmative replies included comments pointing out that the projects introduced the student to the available computing facilities, made the convergence concepts "come alive," illustrated the theory with "physical proof," and made it possible for students to "experiment with mathematics." One of the two negative replies was accompanied by the comment that the "computer cannot explain something" about which a student is already confused.

The response to question two constitutes the reason for which this article has been written. Of the twenty-seven students responding, none thought that canned programs should be used in the course. Twenty-four agreed with the approach that was actually taken, and three would have preferred to have written programs from scratch without the benefit of sample programs. Comments here pointed out that canned programs would not get the student as involved with the mathematics associated with the programs, and that writing all programs from scratch would have made too great a demand on both class and study time.

In response to question three, most students simply reaffirmed their satisfaction with the present approach. Several commented that they would like to have had more programs, a good number felt that the programs should be made mandatory of all students, and several made the suggestion that one or two of the last programs should be written from scratch by the students themselves.

The projects resulted in very little out of class demand on the instructor's time. The diagnostic features of the PL/1 language are quite extensive. Thus a student, armed with a sample of a correct program, can usually deduce the source of his difficulties from the diagnostic error messages and from a model of what he should have entered. Also, as is to be expected, several of the students had had some prior experience with computing. These students are generally delighted to practice their skills by helping classmates with their programming difficulties, and we suspect that most of the necessary assistance was obtained in this way.

Conclusions

It is this author's belief that in most colleges and universities good use can be made of the available computing facilities by integrating their use into the traditional presentation of the introductory calculus. Students need to know that their computing facilities can now be approached in much the same way as their hand calculators—one simply

needs to determine what the available facilities are, how to gain access to them, and what the rudiments of an appropriate language happen to be. Indeed, the cumbersome approach of separate courses or lengthy tutorials in programming have probably served to insure that some of our calculus students will take pains to avoid coming into contact with computing, since it has been made to appear so difficult.

As a final note it is interesting to observe the case of Dartmouth, where the concept of time-sharing was first introduced (1964) and where Kemeny produced the BASIC language. In the report [15] Stephen Garland notes that Dartmouth originally began with the supplemental approach to calculus and computing, then developed a more fully integrated approach, but has now come around to a state of more informal usage:

> ...students do as much or as little (programming) as they wish, with their activity ranging from using programs written by instructors to writing their own simple programs to developing a full-fledged interest in computing. Perhaps this last approach indicates that computing has finally come of age at Dartmouth; students and faculty expect to use computing just as they expect to use any other intellectual tool, but no longer feel compelled to call attention to the tool every time they use it.

References

1. Caldwell, J. L. and Polley, Douglas, "Computer Assisted Tutorials in College Mathematics," American Mathematical Monthly 82 (1975) 846-848.
2. Conway and Gries, *An Introduction to Programming*, 2nd Edition, Winthrop, New York, 1975.
3. Garfunkel, Solomon, "A Laboratory and Computer Based Approach to Calculus," American Mathematical Monthly 79 (1972) 282-290.
4. Greiser, Tim, *Go—A RAX Primer*, Boston University Computer Center Publication.
5. Hethcote, H. W., and Schaffer, A. J., "A Computer Laboratory Course for Calculus and Linear Algebra," American Mathematical Monthly 79 (1972) 290-293.
6. Lax, Peter, Burnstein, Samuel, and Lax, Anneli, *Calculus with Applications and Computing*, Springer, New York, 1976.
7. Menzin, Margaret S., *Use of Canned Computer Programs in Freshman Calculus*, American Mathematical Monthly 83 (1976) 375-378.
8. Miles, E. P. Jr., "Calculus and the Computer, the CRICISAM Calculus Project—Past, Present and Portent," American Mathematical Monthly 78 (1971) 284-291.
9. Osofsky, Barbara, "Small Calculators for the Mathematician," Notices of the American Mathematical Society 23 (1976) 422-426.
10. Pollach and Sterling, *A Guide to PL/1*, 2nd Edition, Holt, New York, 1976.
11. O'Loughlin, Thomas, "Using the Electronic Programmable Calculator in Calculus Instruction," American Mathematical Monthly 83 (1976) 281-283.
12. Thesing, G. L. and Wood, C. A., "Using the Computer as a Discovery Tool in Calculus," American Mathematical Monthly 81 (1975) 163-168.
13. Wegner, Peter, "A View of Computer Science Education," American Mathematical Monthly 79 (1972) 168-179.
14. Williams, H. E. and DeBoer, Delmer, "Teaching a Computer-Oriented Laboratory Course for Ordinary Differential Equations," American Mathematical Monthly 80 (1973) 933-937.
15. "Calculus and Computing: The Influence of Computing on Mathematical Research and Education," Joint publication of the American Mathematical Society and the Mathematical Association of America, 1975.
16. "Calculus with Computers," report of CUPM, 1969.
17. "Recommendations on Undergraduate Mathematics Courses Involving Computing," report of CUPM, 1972.